

### Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application. The following listing provides the amended claims with the amendments marked with deleted material crossed out and new material underlined to show the changes made.

1. (Currently Amended) For a router that uses a set of partitioning lines to partition a region of a design layout into a plurality of sub-regions, a method of pre-computing routes for nets, the method comprising:

a) for each particular set of potential sub-regions, identifying a set of routes that traverse the particular set of potential sub-regions;

b) for each ~~particular route~~ identified route for each particular set of sub-regions, determining whether the ~~particular~~ identified route is stored in a storage structure; and when the ~~particular~~ identified route is not stored in the storage structure, storing the ~~particular~~ identified route in the storage structure.

2. (Original) The method of claim 1, wherein some of the identified routes have diagonal route edges.

3. (Currently Amended) The method of claim 2, wherein a first route is identified for a first set of potential ~~sub-region~~ sub-regions, and the first route is stored in the storage structure, the method further comprising associating the first route with the first set of potential sub-regions.

4. (Original) The method of claim 2 further comprising: when a first route that is identified for a first set of sub-regions is already stored in the storage structure, associating the first route with the first set of potential sub-regions.

5. (Currently Amended) The method of claim 2, wherein determining whether the particular identified routes are stored in the storage structure comprises:

using a binary tree to sort the stored routes; and

traversing the binary tree to determine whether routes are stored in the storage structure.

6. (Currently Amended) For an electronic design automation ("EDA") router that routes nets within a region of an integrated-circuit layout, a method of pre-computing routes, the method comprising:

a) defining a set of partitioning lines for partitioning the region into a plurality of sub-regions;

b) for a first set of sub-regions,

identifying a first set of routes that connect the first set of sub-regions;

storing the first set of routes in a storage structure; and

establishing a relationship between the first set of routes and the first set of sub-regions;

and

c) for a second set of sub-regions,

identifying a second set of routes that connect the second set of sub-regions;

determining whether each route in the second set is stored in the storage structure;

when a new route in the second set is not stored in the storage structure, storing the new route in the storage structure and establishing a relationship between the new route and the second set of sub-regions; and

when an repeating route in the second set is stored in the storage structure, establishing a relationship between the repeating route and the second set of sub-regions.

7. (Original) The method of claim 6, wherein some of the identified routes have diagonal route edges.

8. (Currently Amended) The method of claim [[1]] 6, wherein determining whether particular routes are stored in the storage structure comprises:

using a binary tree to sort the stored routes; and

traversing the binary tree to determine whether routes are stored in the storage structure.

9. (Original) The method of claim 8, wherein each route is specified by a bitstring having a plurality of bits, wherein traversing the binary tree comprises traversing the tree based on the bits in the bitstrings of the routes.

10. (Currently Amended) The method of claim 9, wherein the binary tree includes nodes and each node includes either zero or two branches,

wherein each node corresponds to a particular bit in the bitstrings, and

wherein traversing the binary tree based on the route bitstrings comprises identifying, for a particular route bitstring, the branch to select at a particular node based on the value of the route bitstring's bit that corresponds to the particular node.

11. (Original) The method of claim 9 wherein a plurality of paths exist between the sub-regions defined by the set of partitioning lines, wherein each route's bitstring bits represent the paths used by the route.

12. (Original) The method of claim 11, wherein a plurality of the paths are diagonal paths, and wherein some of the routes traverse some of the diagonal paths.

13. (Original) The method of claim 9 wherein a plurality of inter-sub-regions exist between the sub-regions defined by the set of partitioning lines, wherein each route's bitstring bits represent the inter-sub-regions intersected by the route.

14. (Original) The method of claim 13, wherein a plurality of the inter-sub-regions are diagonal inter-sub-regions, and wherein some of the routes intersect some of the diagonal inter-sub-regions.

15. (Original) The method of claim 6, wherein establishing a relationship between each particular set of sub-regions and each set of routes for the particular set of sub-regions comprises:

for each set of sub-regions, storing a set of references to routes,

wherein each stored reference set includes one or more references to stored routes.

16. (Original) The method of claim 6, wherein the sets of routes for some set of sub-regions includes only one route, while the sets of routes for other set of sub-regions includes more than one route.